REMARKS

The Office Action dated February 10, 2005 has been received and carefully considered. In this response, claims 1, 5, 7-11, 13, 14, 16, 18-25, 27, 28, 30, 47-49 and 54 have been amended to remove unnecessary "step for" language and to correct various informalities and claims 55-62 have been canceled without prejudice. These amendments do not narrow the scope of the claims. Support for the amendments to the claims may be found in the specification and figures as originally filed. Entry thereof and reconsideration of the outstanding objections and rejections therefore is respectfully requested.

Indefinite Rejection of Claim 48

At page 2 of the Office Action, claim 48 was rejected under 35 U.S.C. Section 112, second paragraph, as being indefinite. In view of the Examiner's remarks, the Applicants have amended claim 48 to more clearly recite the features of wherein the second software driver is less protected from unauthorized access than the first software driver. Withdrawal of the indefinite rejection of claim 48 therefore is respectfully requested.

Anticipation Rejection of Claims 1-3, 8-13, 16, 18, 24-26, 31-33, 38, 39, 47-51 and 54

At page 3 of the Office Action, claims 1-3, 8-13, 16, 18, 24-26, 31-33, 38, 39, 47-51 and 54 were rejected under 35 U.S.C. Section 102(e) as being anticipated by Ciacelli (U.S. Patent No. 6,236,727). This rejection is respectfully traversed.

Claim 1, from which claims 2, 3, 8-13 and 16 depend, recites the features of sending a first encrypted routine of a software driver to a peripheral device, decrypting, at the peripheral device, the first encrypted routine to generate a plaintext routine, and providing the plaintext routine to the software driver. The Examiner asserts that the passages of Ciacelli at col. 5, lines 43-48 and col. 5, lines 54-55 disclose these features. *See Office Action*, p. 3. For ease of reference, Figure 1 of Ciacelli and a passage of Ciacelli including the cited passages of Ciacelli is reproduced below:

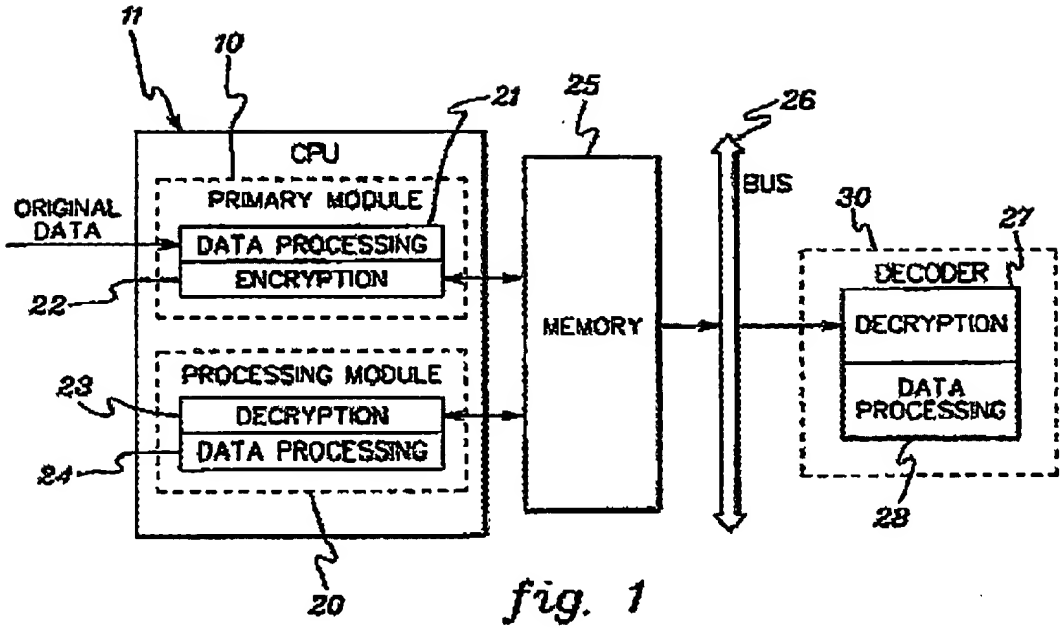


fig. 1

Ciacelli, Figure 1

Alternatively, encryption module 22 and decryption module 23 (or decryption device 27) can be predefined at the design stage to include a resident encryption/decryption routine. Before encryption, module 22 would decide on an actual encryption and decryption algorithm pair to be used. Module 22 would use the resident encryption algorithm to encrypt the actual decryption routine of the selected algorithm pair to be used by the decryption module 23 and/or decryption device 27. The encryption module 22 then transmits the encrypted version of the actual decryption algorithm to module 23 and/or device 27.

Upon receipt of this information, the decryption module 23 and/or device 27 employs the resident decryption algorithm to decrypt the downloaded algorithm. Module 23 then uses the descrambled decryption algorithm as a procedure call, while device 27 could load the algorithm into a programmable circuit within device 27. After completing downloading of the actual decryption algorithm, module 22 uses the actual encryption algorithm to encrypt the data, and module 23 and/or device 27 employs the downloaded decryption routine to decrypt the data. If an update of the encryption/decryption routine is desired, then a different encryption/decryption algorithm pair is selected and encryption module 22 downloads the corresponding decryption algorithm into the decryption module 23 and/or decryption device 27.

After decryption is performed, the receiving data processing module 24 and/or device 28 performs any required data processing, such as MPEG decoding of a clear, compressed video/audio data signal.

Ciacelli, col. 5, lines 35 – 60.

The Applicants respectfully submit that the relied-upon passages of Ciacelli fail to disclose decrypting a first encrypted routine *from a software driver* at a peripheral device to generate a plaintext routine and providing the plaintext routine *to the software driver*. As Figures 2 and 3 and the above-cited passage illustrate, Ciacelli discloses that the encryption module 22 uses an encryption algorithm to encrypt a decryption routine and transmits the encrypted version of the decryption routine to the decryption module 23 and/or the device 27. *See Id.* The decryption module 23 and/or the device 27 then use a decryption algorithm corresponding to the encryption algorithm to decrypt the encrypted version of the decryption algorithm. *See Id.* The descrambled decryption routine then may be used by the decryption module 23 and/or the device 27 to decrypt data that has been encrypted by the encryption module 22. *See Id.*

From the Examiner's reliance on the above-cited passages of Ciacelli, it appears that the Examiner considers the encryption module 22 as the same as or equivalent to the software driver of claim 1 and either the decryption module 23 or the device 27 as the same as or equivalent to the peripheral device of claim 1. Assuming, *arguendo*, that these equivalencies are proper, it is respectfully submitted that Ciacelli fails to disclose that the descrambled decryption routine is provided from the decryption module 23 or device 27 to the encryption module 22 and therefore fails to disclose the features of providing the plaintext routine to the software driver as recited by claim 1. As noted above, the encryption module 22 encrypts a decryption algorithm and provides the encrypted version of the decryption algorithm to the decryption module 23/device 27, which the decryption module 23/device 27 decrypts and uses the descrambled decryption algorithm to decrypt encrypted data provided by the encryption module 22. *See Id.* Neither the cited passage nor any other passage of Ciacelli discloses that the descrambled decryption algorithm is provided from the decryption module 23/device 27 to the encryption module 22. In fact, there would be no need for the decryption module 23/device 27 to provide the descrambled decryption algorithm to the encryption module 22 as it was the encryption module 22 that encrypted the decryption algorithm in the first place. *See Id.*, col. 5, lines 40-43 ("Module 22 would use the resident encryption algorithm to encrypt the actual decryption routine of the selected algorithm pair to be used by the decryption module and/or device 27"). Accordingly, Ciacelli fails to disclose at least the features of providing the plaintext routine to the software

driver as recited by claim 1. The Office Action therefore fails to establish that Ciacelli discloses each and every feature of claim 1, as well as each and every feature of claims 2, 3, 8, 9-13 and 16 at least by virtue of their dependency from claim 1. Moreover, these claims recite additional features not disclosed by Ciacelli.

To illustrate, claim 2 recites the additional features of wherein the first encrypted routine is an encrypted version of an encryption routine. The Examiner asserts that the passages of Ciacelli at col. 5, lines 43-50 (reproduced above), col. 6, lines 54-60 and col. 9, lines 6-14 disclose these features. In particular, the Examiner asserts that "re-encryption is employed after CSS decryption and thereby not only the decryption algorithm but also the encryption algorithm are needed with the encrypted version (Ciacelli: see for example: Column 9 Line 6-14)." *Office Action*, p. 4. For ease of reference, the cited passage of Ciacelli at col. 9, lines 6-14 is reproduced below:

> FIG. 4 depicts a further embodiment of processing in accordance with the present invention. In this embodiment, rescrambling of the data stream is employed after CSS decryption, along with subsequent descrambling of the re-encrypted stream prior to decompression decoding in a decoder chip. The processings described are preferably accomplished within on-chip microcode.

Ciacelli, col. 6, lines 54-60.

> 10. The apparatus of claim 8, wherein said means for selecting comprises means for selecting said encryption/decryption algorithm pair from a plurality of encryption/decryption algorithm pairs at said re-encryption means and said decryption means, and wherein said means for selecting comprises means for noticing the decryption means which decryption algorithm of said plurality of encryption/decryption algorithm pairs corresponds with an encryption algorithm employed by said re-encryption means.

Ciacelli, col. 9, lines 6-14.

Considering, again, the Examiner apparent equivalencies between the encryption module 22 and the software driver of claim 1 and between the decryption module 23/device 27 and the peripheral device of claim 1, it is noted that the passages of Ciacelli at col. 6, lines 54-60 and col. 9, lines 6-14 do not disclose that an encrypted version of an encryption routine is provided to any component, much less from the encryption module 22 to the decryption module 23/device 27. Instead, these passages merely provide that "rescrambling of the data stream is employed after

CSS decryption." As noted above, the passage of Ciacelli at col. 5, lines 35-60 discloses only that a decryption routine is encrypted and the encrypted version of the decryption routine is provided to the decryption module 23/driver 27. Ciacelli does not disclose in any manner that an encrypted version of an encryption routine is sent from a software driver to a peripheral device, decrypted at the peripheral device to generate a plaintext routine (the decrypted version of the encryption routine) and that the plaintext routine (the decrypted version of the encryption routine) is provided to the software driver as recited by claim 3.

As another example, claim 10 recites the additional features of encrypting, at the peripheral device, the plaintext routine to generate a second encrypted routine, where the second encrypted routine is a version of the first encrypted routine and providing the second encrypted routine to the software driver. With respect to these features, the Examiner cites the passages of Ciacelli at col. 7, lines 16-22 and 58-65 and col. 9, lines 6-14, and asserts that "Ciacelli teaches (a) clear data is "never" resident in an accessible computer system such as memory buffer to inhibit theft . . . and thereby the plaintext routine must be resident in the encrypted form, (b) re-encryption mechanisms through multiple 'devices'." *Office Action*, p. 5. As a first issue, one of ordinary skill in the art will appreciate that the term "plaintext" indicates unencrypted information and therefore the Examiner's assertion that "the plaintext routine must be resident in the encrypted form" is a contradiction in terms. Moreover, Ciacelli teaches that "clear data . . . is never resident *in an accessible computer system structure*, such as a host memory buffer or system bus . . . ." *Ciacelli*, col. 7, lines 17-21 (emphasis added). As Ciacelli teaches that the encryption module 22 and the decryption module 23 are part of software modules 10 and 20, respectively, executed by a CPU and the device 27 is part of a processing unit hardware device, one of ordinary skill in the art will appreciate that the encryption module 22, the decryption module 23 and the device 27 are not "an accessible computer system" like a host memory buffer or system bus. Accordingly, Ciacelli provides no indication that clear data at these components "is never resident" as they are not readily accessible computer system structures. As a second issue, even if Ciacelli disclosed storing a re-encrypted version of the decrypted decryption algorithm (which Ciacelli fails to do), Ciacelli provides no disclose that such a re-encrypted version is provided to the encryption module 22 (which, as noted above, the Examiner appears to equate to the software driver of claim 1). Accordingly, Ciacelli fails to disclose that the plaintext

routine is encrypted to generate a second encrypted routine and that the second encrypted routine is provided to the software driver as recited by claim 10.

As yet another example, claim 9 recites the additional features of removing the plaintext routine from the software driver. The Examiner cites the passage of Ciacelli at col. 7, line 16-21 as disclosing these features. However, this passage provides merely provides that "clear data . . . is never resident *in an accessible computer system structure*, such as a host memory buffer or system bus . . . " and, as noted above, one of ordinary skill in the art will appreciate that the encryption module 22, the decryption module 23 and the device 27 are not "accessible computer system structures." Moreover, even if it is assumed that they are "accessible computer system structures," Ciacelli teaches that clear data "is never resident" in "accessible computer system structures," so such a structure could "never" store a plaintext routine. As such a structure could "never" store a plaintext routine, the plaintext would not be available for removal from such a structure. Accordingly, Ciacelli fails to disclose the features of removing the plaintext routine as recited by claim 9.

As an additional example, claim 16 recites the features of wherein providing includes storing the plaintext routine in a location in memory accessible by the software driver, and where the location in memory is known to the software driver. The Examiner cites the passages of Ciacelli at col. 5, lines 45-55 and col. 6, lines 54-60 as disclosing these features and further reasons that "the memory location storing the decrypted 'encryption/decryption routine' must be known to the software driver so that the re-encryption/decryption function can be preformed and executed accordingly to re-encrypt/decrypt the data." *Office Action*, p. 7. As noted above, Ciacelli fails to disclose that the decrypted decryption algorithm (which the Examiner appears to consider equivalent to the plaintext routine) is provided or made available to the encryption module 22 (which the Examiner appears to consider equivalent to the software driver) in any manner. Moreover, it is respectfully submitted that neither the cited passages nor any other passage of Ciacelli disclose that the decrypted decryption algorithm (which the Examiner appears to consider equivalent to the plaintext routine) is stored in memory in any manner. In fact, such an interpretation would be contrary to the teachings of Ciacelli because, as noted by the Examiner and as discussed above, Ciacelli teaches that "clear data . . . is never resident in an accessible computer system structure, such as a host memory buffer or system bus. . ." and

because memory would qualify as an "accessible computer system structure", a plaintext routine (which is analogous to "clear data") would "never" be "resident" in memory. *See Ciacelli*, col. 7, lines 18-21. Accordingly, Ciacelli fails to disclose the features of storing the plaintext routine in memory accessible by the software driver, and where the location in memory is known to the software driver as recited by claim 16.

In view of the foregoing, it is respectfully submitted that the Office Action fails to establish that Ciacelli anticipates claims 1-3, 8-13 and 16.

Claim 17, from which claims 18 and 24-26 depend, recites the features of sending a first encrypted routine of a software driver to a graphics chip, wherein the software driver is to interface with the graphics chip, and where the first encrypted routine is an encrypted version of an encrypted routine. Claim 17 further recites the features of decrypting, at the graphics chip, the first encrypted routine to generate a plaintext routine, wherein the plaintext routine is a version of the encryption routine, and storing the plaintext routine in memory in a location known to the software driver.[1] As acknowledged by the Examiner, Ciacelli does not disclose a graphics chip. Moreover, as discussed above, Ciacelli does not disclose sending an encrypted version of an encryption routine. Accordingly, Ciacelli fails to disclose sending a first encrypted routine of a software driver to a graphics chip, where the first encrypted routine is an encrypted version of an encrypted routine as recited by claim 17. Moreover, as discussed above, Ciacelli fails to disclose that a plaintext routine is made available to a software driver in any manner. Accordingly, Ciacelli fails to disclose storing the plaintext routine in a memory location known to the software driver as recited by claim 17. The Office Action therefore fails to establish that Ciacelli discloses each and every feature of claim 17, as well as each and every feature of claims 18 and 24-26 at least by virtue of their dependency from claim 17. Moreover, these claims recite additional features not disclosed by Ciacelli.

[1] The Office Action rejects claim 17 under 35 U.S.C. Section 103 in view of Ciacelli, whereas claims 18 and 24-26, which depend from claim 17, are rejected under 35 U.S.C. Section 102 in view of Ciacelli. It will be appreciated that the dependent claims 18 and 24-26 incorporate all of the features of the parent claim 17 by virtue of their dependency from claim 17. Accordingly, the Applicants submit that an anticipation rejection of dependent claims 18 and 24-26 is improper when the parent claim 17 is subject only to an obviousness rejection using the same reference.

For example, claim 24 recites the additional features of encrypting, at the graphics chip, the plaintext routine to generate a second encrypted routine, and storing the second encrypted routine in memory in a location known to the software driver. As noted above with respect to claim 10, Ciacelli fails to disclose these features.

Claim 31, from which claims 32, 33, 38 and 39 depend, recites the features of a peripheral device to decrypt a first encrypted routine and generate a plaintext routine and a software driver including a program of instructions to manipulate a processor to send the first encrypted routine of the software driver to the peripheral device and execute the plaintext routine. As noted above with respect to claims 1 and 17, Ciacelli fails to disclose that a plaintext routine generated from the decryption of an encrypted routine is provided to a software driver in any manner. Ciacelli therefore necessarily fails to disclose the features of a peripheral device to decrypt a first encrypted routine and generate a plaintext routine and a software driver including a program of instructions to manipulate a processor to execute the plaintext routine as recited by claim 31. Accordingly, Ciacelli fails to disclose each and every feature of claim 31, as well as each and every feature of claims 32, 33, 38 and 39 at least by virtue of their dependency from claim 31. Moreover, these claims recite additional features not disclosed by Ciacelli.

For example, claim 39 recites the additional features of wherein the peripheral device includes a hardware component to encrypt the plaintext routine to generate a second encrypted routine. As noted above with respect to claims 10 and 24, Ciacelli fails to disclose that the decryption module 23/device 27 (which the Examiner appears to consider equivalent to the peripheral device of claim 31) performs an encryption operation in any manner, much less encrypting the plaintext routine to generate a second encrypted routine as recited by claim 39.

Claim 47, from which claim 48 depends, recites the features of sending a first encrypted routine of a first software driver to a peripheral device, decrypting, at the peripheral device, the first encrypted routine to generate a plaintext routine, and providing the plaintext routine to a second software driver. As noted above, Ciacelli does not disclose that the plaintext routine is provided to a software driver in any manner, so Ciacelli necessarily fails to disclose that the plaintext routine is provided to a second software driver as recited by claim 47. Ciacelli therefore fails to disclose each and every feature of claim 47, as well as each and every feature of

claim 48 at least by virtue of its dependency from claim 47. Moreover, claim 48 recites additional features not disclosed by Ciacelli.

To illustrate, claim 48 presently recites the additional features of wherein the second software driver is less protected from unauthorized access than the first software driver. Although the Examiner cites the passages of Ciacelli at col. 5, lines 43-45 and col. 6, lines 54-60 in rejecting claim 48, these passages provide no mention of one software driver being less protected from unauthorized access that another software driver, much less that the second software driver receiving a plaintext routine is less protected from unauthorized access than a first software driver that provides an encrypted routine from which the plaintext routine is generated as provided by claim 48.

Claim 49, from which claims 50, 51 and 54 depend, recites the features of sending a first encrypted data associated with an application to a peripheral device, decrypting, at the peripheral device, the first encrypted data to generate a plaintext data, and providing the plaintext data to the application. As similarly noted above with respect to claims 1 and 31, Ciacelli fails to disclose that the decryption module 23/device 27 (which the Examiner appears to consider equivalent to the peripheral device of claim 49) provides any plaintext data to the encryption module 22 (which the Examiner appears to consider equivalent to the application of claim 49), where the plaintext data was generated by decrypting, at the peripheral device, encrypted data associated with the application. Accordingly, Ciacelli fails to disclose the features of providing the plaintext data to the application of claim 49. Ciacelli therefore fails to disclose each and every feature of claim 49, as well as each and every feature of claims 50, 51 and 54 at least by virtue of their dependency from claim 49.

For example, claim 54 recites the additional features of encrypting, at the peripheral device, the plaintext data to generate a second encrypted data. As noted above, Ciacelli fails to disclose that the decryption module 23/device 27 (which the Examiner appears to consider equivalent to the peripheral device of claim 49) performs an encryption operation in any manner.

In view of the foregoing, the Applicants respectfully submit that the anticipation rejection of claims 1-3, 8-13, 16, 18, 24-26, 31-33, 38, 39, 47-51 and 54 is improper at the withdrawal of this rejection therefore is respectfully requested.

Obviousness Rejections of Claims 4-7, 14, 15, 17, 19-23, 30, 34-38, 40-46, 52, 53 and 55-62

At page 8 of the Office Action, claims 4-7, 17, 19-23, 30, 34-38, 40-46, 52, 53 and 55-62 were rejected under 35 U.S.C. Section 103(a) as being unpatentable over Ciacelli in view of Freeman (U.S. Pat. App. Pub. No. 2002/0129374). At page 15 of the Office Action, claims 14 and 15 were rejected under 35 U.S.C. Section 103(a) as being unpatentable over Ciacelli in view of Wilson (U.S. Patent No. 4,520,232). These rejections are respectfully traversed.

Claim 1, from which claims 4-7, 14 and 15 depend, recites the features of sending a first encrypted routine of a software driver to a peripheral device, decrypting, at the peripheral device, the first encrypted routine to generate a plaintext routine, and providing the plaintext routine to the software driver. Claim 17, from which claims 19-23 and 30 depend, recites the features of sending a first encrypted routine of a software driver to a graphics chip, wherein the software driver is to interface with the graphics chip, and where the first encrypted routine is an encrypted version of an encrypted routine. Claim 49, from which claims 52 and 53 depend, recites the features of sending a first encrypted data associated with an application to a peripheral device, decrypting, at the peripheral device, the first encrypted data to generate a plaintext data, and providing the plaintext data to the application. As noted above, Ciacelli fails to disclose or suggest these features of claims 1, 17 and 49. Moreover, the Office Action does not assert that Freeman or Wilson discloses or suggests these features, nor, in fact, do Freeman or Wilson disclose or suggest these features. Instead, the Examiner appears to rely on Freeman and Wilson mainly as showing a graphics chip and the use of a map as an encryption key, respectively. *See Office Action*, pp. 9-15. Accordingly, the proposed combinations of Ciacelli, Freeman and Wilson fail to disclose or suggest each and every feature of claims 1, 17 and 49, as well as each and every feature of claims 4-7, 14, 15, 19-23, 30, 52 and 53 at least by virtue of their dependency from one of claims 1, 17 or 49. Moreover, these claims recite additional features neither disclosed nor suggested by Ciacelli, Freeman or Wilson.

For example, claim 15 recites the additional features of wherein decrypting includes using a map as a decryption key, wherein the map includes a texture map. The Examiner cites the passages of Wilson at col. 2, lines 12-24 and col. 1, lines 28-34 in rejecting claim 15 and asserts that "the matrix [disclosed by Wilson at col. 2, lines 12-24] is qualified as a two-

dimensional texture map." *Office Action*, p. 16. However, contrary to the Examiner's assertions, the matrix of Wilson is not the same or equivalent to a texture map. Wilson merely provides that an encryption or decryption key may be stored in the form of a binary matrix. That the binary matrix of Wilson and a texture map may have the same organization representation (i.e., values arranged in a matrix pattern) does not suggest that the binary matrix of Wilson may be substituted as a texture map or vice versa. Moreover, with respect to the binary matrix, Wilson teaches that "[s]ignificantly, no row of the matrix can be all zeros, and the modulo 2 sum of any combination of all rows cannot be equal to the binary number representative of any row. The same is true for columns." *Wilson*, col. 2, lines 13-16. One of ordinary skill in the art will appreciate that texture maps are not subject to the requirements of the binary map of Wilson and therefore the binary map of Wilson is not the same or equivalent to the features of using a texture map as a decryption key as provided by claim 15.

As another example, claim 23 recites the additional features of removing the plaintext routine and claim 30 recites the additional features of storing the plaintext routine in a location in memory accessible by the software driver. As noted above with reference to claims 10 and 17, respectively, Ciacelli fails to disclose or suggest these features, as do Freeman and Wilson. Moreover, as noted above with respect to claim 17, Ciacelli teaches away from the features of storing the plaintext routine in a location in memory accessible by the software driver as recited by claim 30.

Claim 40, from which claims 41-46 depend, recites the features of a first interface to receive a first encrypted routine of a software driver, a first hardware component to decrypt the first encrypted routine received by said interface and generate a plaintext routine, and a second interface to output the plaintext routine for use by said software driver. As discussed above, Ciacelli fails to disclose decrypting an encrypted routine of a software driver at a hardware component to generate a plaintext routine and providing the plaintext routine for use by the software driver. Accordingly, Ciacelli, as well as Freeman, fails to disclose or suggest a first hardware component to decrypt the first encrypted routine of a software driver and a second interface to output the plaintext routine for use by said driver as recited by claim 40. Accordingly, the proposed combination of Ciacelli and Freeman fails to disclose or suggest each and every feature of claim 40, as well as each and every feature of claims 41-46 at least by virtue

of their dependency from claim 40. Moreover, these claims recite additional features neither disclosed nor suggested by Ciacelli or Freeman.

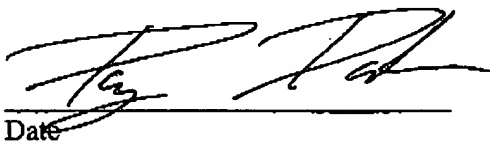
In view of the foregoing, it is respectfully submitted that the obviousness rejections of claims 4-7, 14, 15, 17, 19-23, 30, 34-38, 40-46, 52, 53 and 55-62 are improper and the withdrawal of these rejections therefore is respectfully requested.

Conclusion

It is respectfully submitted that the present application is in condition for allowance and an early indication of the same is courteously solicited. The Examiner is respectfully requested to contact the undersigned by telephone at the below listed telephone number in order to expedite resolution of any issues and to expedite passage of the present application to issue, if any comments, questions, or suggestions arise in connection with the present application.

The Commissioner is hereby authorized to charge any fees that may be required, or credit any overpayment, to Deposit Account Number 50-0441.

Respectfully submitted,



Date

10 May 2005

Ryan S. Davidson, Reg. No. 51,596,
On Behalf Of
J. Gustav Larson, Reg. No. 39,263,
Attorney for Applicant(s)
TOLER, LARSON & ABEL, L.L.P.
5000 Plaza On The Lake, Suite 265
Austin, Texas 78746
(512) 327-5515 (phone) (512) 327-5452 (fax)